

# EXHIBIT FFF

# Plaintiff Intellectual Ventures II Tutorial

*U.S. Patent No. 6,189,271*

Case No. 1:14-cv-04638-AKH  
Hon. Judge Alvin K. Hellerstein  
Hearing Date: March 26, 2015

# The '271 Patent

'271 Multiple  
Compression /  
Decompression  
Patent

## I. U.S. Patent No. 6,189,271

- A. “parallel data compression algorithm”
- B. “parallel data decompression algorithm”

# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

## “parallel data compression algorithm”

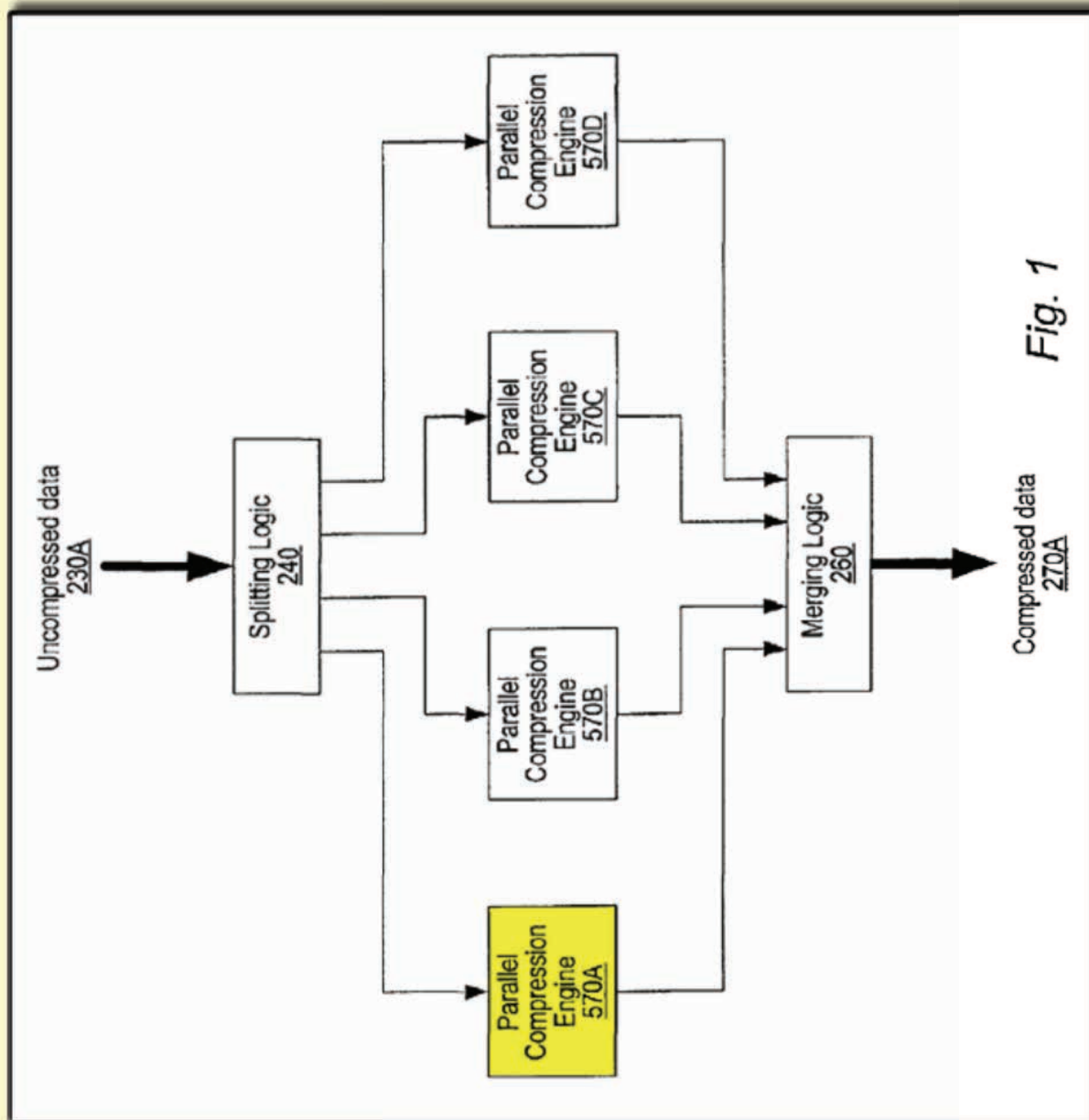
Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel compression engine that compresses more than one data unit at a time	An algorithm that compresses data across multiple processing units at the same time

# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

1. A data compression system comprising:  
a plurality of parallel compression engines, wherein each  
of the plurality of parallel  
compression engines operates independently and imple-  
ments a parallel data  
compression algorithm;  
wherein each of the plurality of parallel compression  
engines is operable to:  
receive a different respective portion of uncompressed  
data; and  
compress the different respective portion of the uncom-  
pressed data using the parallel data  
compression algorithm to produce a respective com-  
pressed portion of the uncompressed data; and  
output the respective compressed portion;  
wherein the plurality of parallel compression engines are  
configured to perform said compression in a parallel  
fashion to produce a plurality of respective compressed  
portions of the uncompressed data.

# '271 Patent: Parallel Data Compression Algorithm



# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel compression engine that compresses more than one data unit at a time	An algorithm that compresses data across multiple processing units at the same time

- ▶ Both Parties' constructions recognize a key statement in the specification (col. 22:13-16):

symbols. Thus the parallel compression algorithm operates on a plurality of symbols at a time. This is different than conventional prior art serial algorithms, which operate in a serial manner on only one symbol at a time. The plurality of

# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel compression engine that compresses more than one data unit at a time	An algorithm that compresses data across multiple processing units at the same time

- ▶ IV's construction closely tracks the specification
- ▶ Specification states that "parallel compression algorithm operates on a plurality of symbols at a time" (Col. 22:14-15)
- ▶ IV's construction requires compressing a plurality of data units at a time
- ▶ A correct claim construction should reflect the inventors' distinguishing over prior art in the specification.
  - ▶ See, e.g., *Kinik Co. v. International Trade Com'n.*, 362 F.3d 1359, 1365 (Fed. Cir. 2004)



# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel compression engine that compresses more than one data unit at a time	An algorithm that compresses data across multiple processing units at the same time

- Citi's construction is flawed because there is no support in claims or specification for "multiple processing units"

## '271 Patent: Parallel Data Compression Algorithm

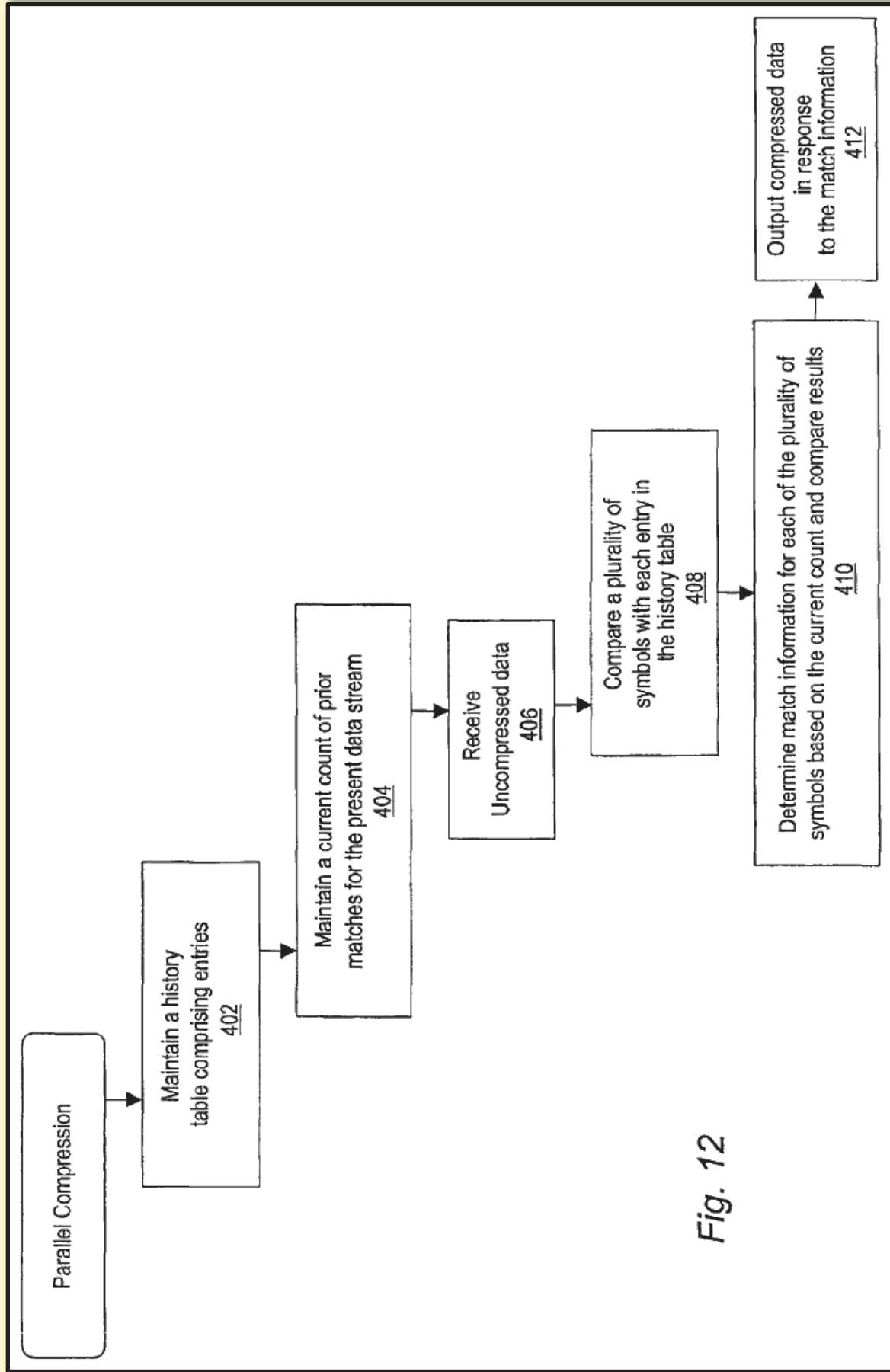


Fig. 12

# '271 Patent: Parallel Data Compression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel compression engine that compresses more than one data unit at a time	An algorithm that compresses data across multiple processing units at the same time

- ▶ The term “data unit” in IV’s construction is supported by the specification
- ▶ “Unit of data” in the specification is representative of data to be operated on (Col. 8:29-31)
- ▶ Examples in the specification:
  - ▶ Pages in memory (Col. 8:29-31)
  - ▶ Symbols (Col. 22:14-15)
  - ▶ Characters (Col. 5:3)
  - ▶ Bytes (Col. 5:3)
  - ▶ Tokens, for decompression (Col. 41:5-7, 33:51)

# The '271 Patent

'271 Multiple  
Compression /  
Decompression  
Patent

## I. U.S. Patent No. 6,189,271

A. “parallel data compression algorithm”

B. “parallel data decompression algorithm”

65. A data decompression system comprising:  
a plurality of parallel decompression engines, wherein each of the plurality of parallel decompression engines operates independently and implements a parallel data decompression algorithm;  
wherein each of the plurality of parallel decompression engines is operable to:  
receive a different respective portion of compressed data; and  
decompress the different respective portion of the compressed data using the parallel data decompression algorithm to produce a respective uncompressed portion of the compressed data; and  
output the respective uncompressed portion;  
wherein the plurality of parallel decompression engines are configured to perform said decompression in a parallel fashion to produce a plurality of respective uncompressed portions of the compressed data.

# '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

Intellectual Ventures' Construction	Defendant's Construction
An algorithm in a parallel decompression engine that decompresses one or more data units at a time	An algorithm that decompresses data across multiple processing units at the same time except in a last decompression cycle

- ▶ Unlike *compression*, there is no definitional statement in specification limiting **decompression** algorithm to operating on *more* than one data unit *at a time*
- ▶ Claims and specification show that **decompression** algorithm operates on **one** or more data units at a time
  - ▶ Each parallel decompression engine still operates in a parallel manner because it operates on its own portion of the data
  - ▶ Serial decompression engines and algorithms operated on all of the data



## '271 Patent: Parallel Data Decompression Algorithm

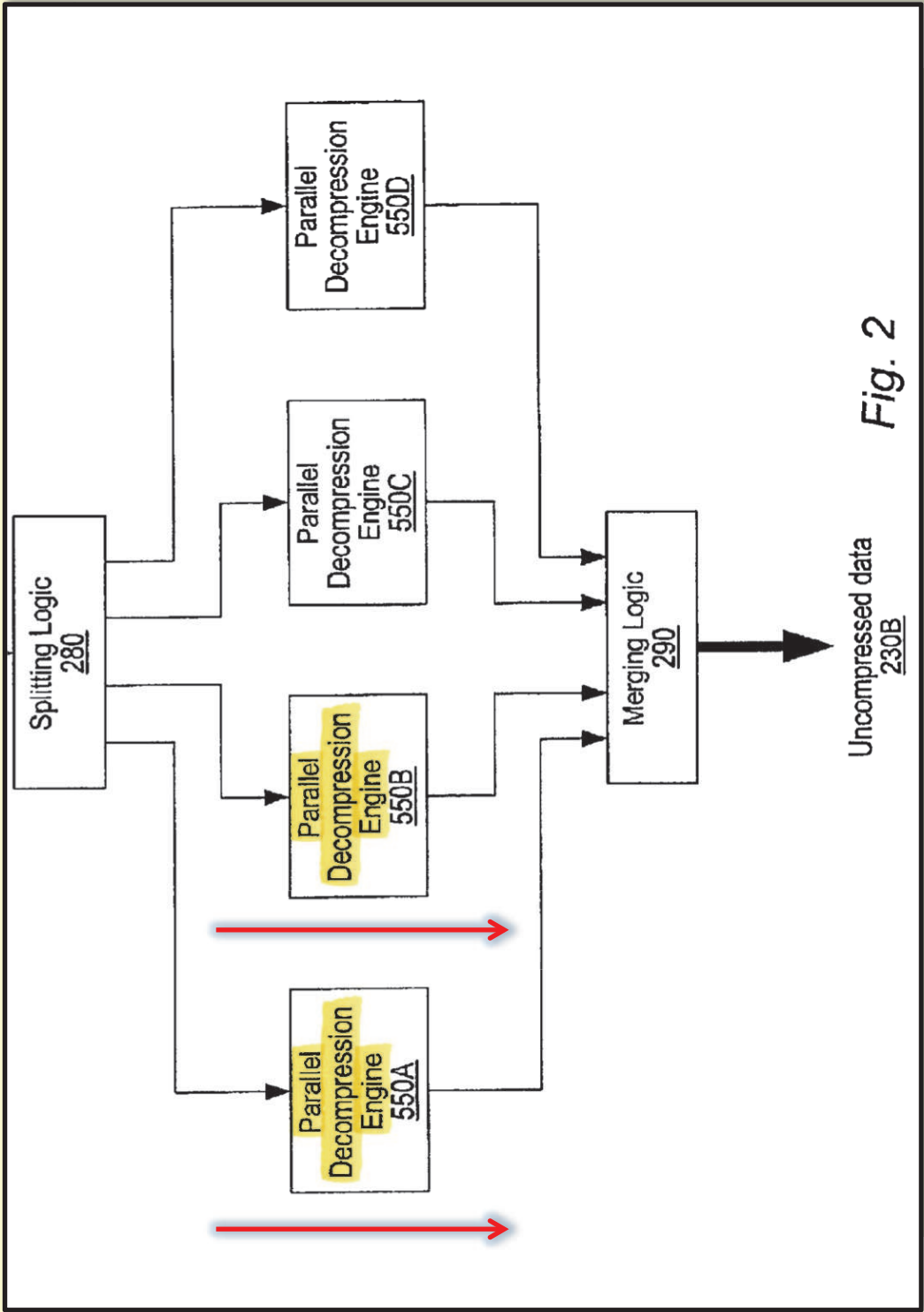
'271 Multiple  
Compression /  
Decompression  
Patent

- ▶ When '271 inventors wanted to specify at-the-same-time operation, they used the word **concurrently**:

66. The data decompression system of claim 65, wherein, in performing said decompression in a parallel fashion, the plurality of parallel decompression engines operate **concurrently** to decompress the different respective portions of the compressed data to produce

73. The data decompression system of claim 72, wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens **concurrently**.

# '271 Patent: Parallel Data Decompression Algorithm



**Concurrent:** at the same time



## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

- ▶ The use of “**concurrently**” in dependent claims means that independent claims are not so limited
- ▶ *Phillips v. AWH Corp.*, 415 F.3d 1303, 1324 (“For example, dependent claim 2 states that the baffles may be ‘oriented with the panel sections disposed at angles for deflecting projectiles such as bullets able to penetrate the steel plates.’ The inclusion of such a specific limitation on the term ‘baffles’ in claim 2 makes it likely that the patentee did not contemplate that the term ‘baffles’ already contained that limitation.”)

## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

- ▶ Citi reads one detail of one embodiment into the claims
- ▶ Specification never limits the decompression algorithm to “multiple processing units at the same time”
- ▶ “[C]ompressed data may be decompressed in a series of cycles, with **one** or more tokens from the compressed data examined and decompressed in parallel **in each cycle**.” (Col. 35:65-36:1)
- ▶ Citi’s construction excludes preferred embodiment
  - ▶ *Vitronics Corp. v. Conceptiontronic, Inc.*, 90 F.3d 1576, 1584 (Fed. Cir. 1996) (A claim construction that excludes a preferred embodiment “is rarely, if ever, correct.”)

# '271 Patent: Parallel Data Decompression Algorithm

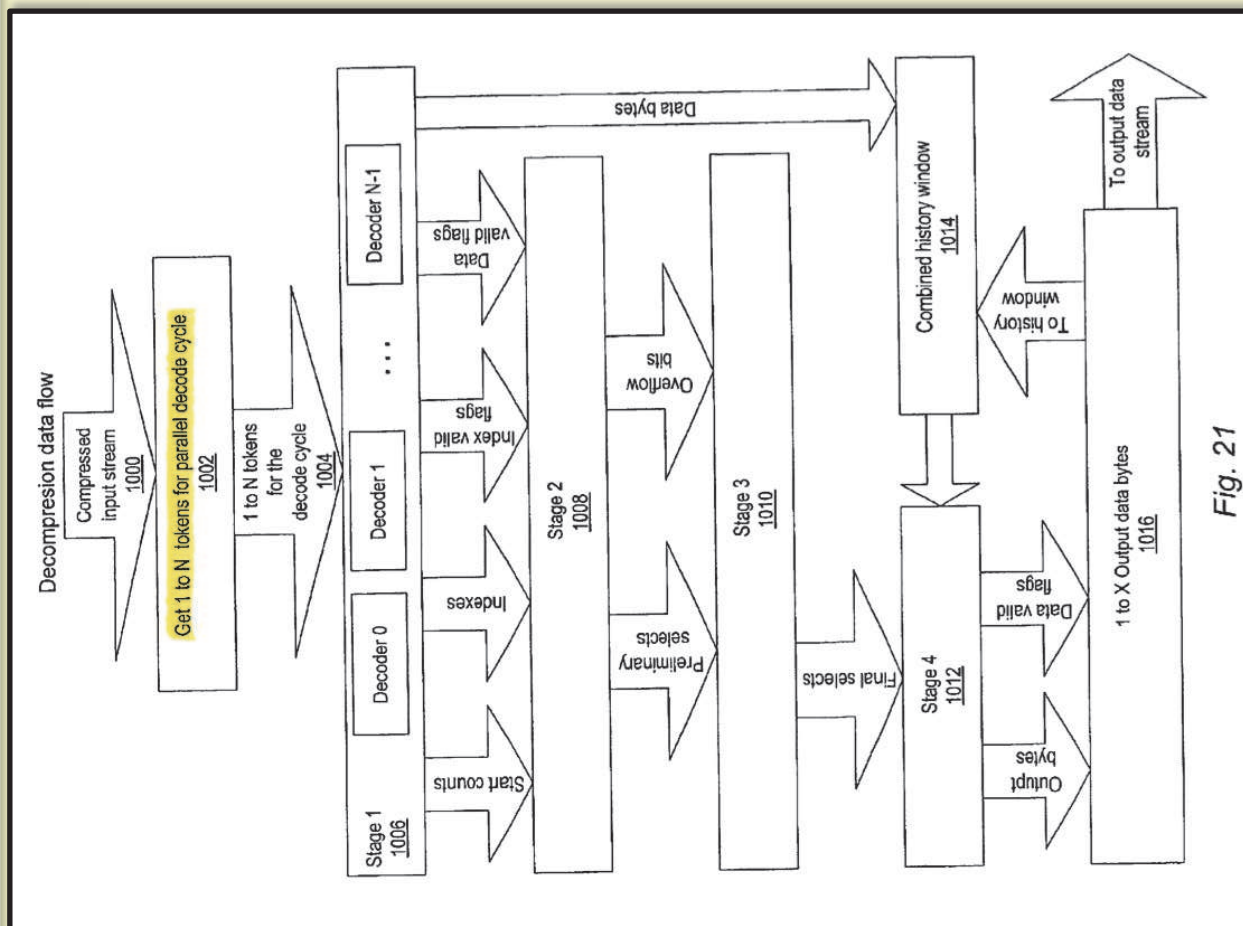


Fig. 21

## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

decompression cycle. Thus, the decompression logic illustrated in FIG. 15 accepts a minimum of one token in a 5 decompression cycle, for example, in a last decompression cycle when only one token is left to decompress. If a token

'271 patent at 37:4-7

## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

pression engine **550**. In one embodiment, one token is **45**  
assigned to one decoder, and one decoder may process one  
token in a decompression cycle. Stage one may include N

'271 patent at 33:45-47

FIG. **23c** expands on block **906** of FIG. **23b**, illustrating  
one embodiment of a method for examining a plurality of  
tokens from the compressed data **900** in parallel. In block  
**908**, one or more tokens to be decompressed in parallel in  
the current decompression cycle may be extracted from the  
compressed data **900**. The tokens may be extracted from the

'271 patent at 36:57-62

## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

FIG. 23*b* illustrates an embodiment of a parallel decompression method performed in one embodiment of the parallel decompression engine 550 of FIG. 23*a*. FIG. 23*b* illustrates that compressed data may be decompressed in a series of cycles, with one or more tokens from the compressed data examined and decompressed in parallel in each cycle. In block 906, the parallel decompression engine may

'271 patent at 35:62-36:1

## '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

data in a cycle. For example, the maximum output of the embodiment of FIG. 15 is 16 bytes per cycle. If a first code being decoded in a first decoder represents more than 16 output bytes, then the first stage 25501 may leave the first code loaded in the first decoder for as many cycles as it take 5 to decompress all of the output bytes represented by the first code. Other codes that may be loaded in the other decoders are not decoded until there are available output data bytes to serve as destinations for the uncompressed symbols to be generated from the tokens. For example, if the first code 10

'271 patent at 29:1-10



# '271 Patent: Parallel Data Decompression Algorithm

'271 Multiple  
Compression /  
Decompression  
Patent

In the preferred embodiment, the decompression engine **550** includes a pipelined, multi-stage design. The pipelined, multi-stage design of the decompression engine **550** enables the substantially simultaneous or concurrent processing of data in each stage. As used herein, the term "decompression

'271 patent at 26:13-17

**73.** The data decompression system of claim **72**, wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens concurrently.

**74.** The data decompression system of **73**, wherein each of the plurality of parallel decompression engines operates in a pipelined fashion; wherein, in examining the plurality of tokens from the different respective portion of the compressed data in parallel, each of the plurality of parallel decompression engines is operable to operate on the plurality of tokens during a single pipeline stage.